



UNIVERSITY OF GOTHENBURG

## Understandings and Implementations of Continuous Delivery

*Bachelor of Science Thesis in the Software Engineering and Management Programme.*

RICKARD BREMER  
JOHAN ERIKSSON

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
Göteborg, Sweden, June 2015

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

#### Understandings and Implementations of Continuous Delivery

Rickard Bremer,  
Johan Eriksson

© Rickard Bremer, June 2015.

© Johan Eriksson, June 2015.

Examiner: Morgan Ericsson

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering  
Göteborg, Sweden June 2015



# Understandings and Implementations of Continuous Delivery

Rickard Bremer and Johan Eriksson

University of Gothenburg, Sweden  
Department of Computer Science and Engineering  
`{gusbremri,guserijobi}@student.gu.se`  
<http://www.gu.se>

**Abstract.** Agile practices are emerging and by adapting Continuous Delivery, agile ideas are turned into practical solutions. The Deployment pipeline begins with Continuous Integration, but it doesn't need to include the deployment of software to target. Understandings and implementations of Continuous Delivery in the industry can differ between organisations. We have conducted a survey study using a questionnaire, to gather data representing a population. We aimed to increase the knowledge in understandings and implementations of Continuous Delivery in the industry. The industry is positive towards Continuous Delivery and do believe that they will benefit from implementing it. But, they don't think their customers are ready to adapt Continuous Deployment at this point. The testing procedure needs to be automated, but some tasks such as testing the user experience is hard to automate. Our research showed us that manual testing is still carried out to a substantial level.

**Keywords:** continuous delivery, continuous deployment, continuous integration, software, agile.

## 1 Introduction

The twelve principles behind Agile Software development aim to act as guidance for developing better software[1]. Principle number one brings up the need to deliver working software frequently. By delivering software to the end user, the organisation or developer will get a final confirmation that they have done the right thing, that the software has a value and that it works properly[2][3]. By doing this more frequently the current path of the development will be continuously validated[4]. A Continuous Delivery process aims to setup an environment where developers can commit code to the master branch whenever he or she have completed a task. If there is something wrong with the code the automated tests will fail and the commit will need to be fixed before replacing the latest version of the software which contains a value for the customer[5][6].

### 1.1 Problem Domain & Motivation

Within industry there is normally a drive to be more efficient when it comes to production, the goal of this is often to get the products out to the market quicker.

By adopting a Continuous Delivery process, parts of the industry believe that they will achieve increased efficiency. The definition of Continuous Delivery can be hard to grasp and hard to communicate while talking with others. Continuous Delivery should not be mistaken for Continuous Deployment, Continuous Deployment is an extension of Continuous Integration[5].

Our interpretations of the three continuous definitions, that we found in the literature, are found in the related work section. We would like to discover how the industry understand the definition of Continuous Delivery and how they would implement such a process. Our perspective is that each organisation will face its own unique challenges and therefore the perspective of what a Continuous Delivery process is, may vary between the organisations. There are some general thoughts that the entire test part in the process need to be automated. The industry might not be able to do so and may have a different perception of this part of the process. Therefore, the study also focused at the testing part of a Continuous Delivery process. We do hope that by collecting data, we can find some clarifications among this data. We may also help organisations or researchers in future implementations or work, to carry out their tasks.

## 1.2 Research Goal & Research Questions

The goal of the survey performed in the course of the Bachelor Thesis is to identify how the industry understands and implements Continuous Delivery and how the industry believe the practice of Continuous Delivery will affect how we test software.

**RQ1 :** How is Continuous Delivery understood in the industry?

Since Continuous Delivery is depending on a working testing procedure, we will not only ask for their opinions about the process, but also look into their current practices regarding the testing procedure, how do they look at it and how is testing implemented in their practices.

**RQ2 :** How does the practice of Continuous Delivery affect the testing procedure within the development cycle?

## 1.3 Scope of the Work

We designed and conducted a survey study, resulting in a questionnaire consisting of three parts. A few background questions, understandings about Continuous Delivery and the last part handles testing in Continuous Delivery. The questionnaire aims to cover obstacles, challenges as well as expectations from different angles. All questions are aimed at finding how the industry interprets and what expectations they do have on a Continuous Delivery process.

The interviews were conducted at eleven organisations with 14 different subjects in and around Gothenburg, Sweden. The organisations had different directions within the field of software development. A part of them worked only with pure software, such as web services, while others used a combination of software and hardware in their products.

#### 1.4 Contributions of the Work

A key finding discovered was that it was important for some organisations to underline and make a clear distinction between Continuous Delivery and Continuous Deployment.

Another key finding was that a majority of the organisations interviewed witnessed that the processes Continuous Integration and Continuous Delivery had a major impact at their look at testing.

#### 1.5 Structure of the Article

The remaining parts of this Bachelor Thesis is structured as follows. Section 2 presents the already conducted work and research within the field. Section 3 describes our methodology for carrying out this survey study. Our results are presented in section 4 and the analysis with a discussion in section 5. Finally the thesis is concluded in section 6 where also future work is presented.

## 2 Related Work

The Continuous Deployment pipeline has been clearly described in the book Continuous Delivery[5]. A Continuous Deployment pipeline is the set of tools, which enables the workflow to deliver source code from the version control system through the build system and the tests to production in a continuous matter, preferably would be that all steps are automated. The code shall be ready for being deployed to target after passing through the Continuous Deployment pipeline.

The first step in the Continuous Deployment pipeline is Continuous Integration, which aims to continuously integrate source code to the main branch[7]. Each commit shall result in a working build of the software. Continuous Integration is essential to be able to establish a Continuous Delivery process. But even Continuous Integration can be a challenging task to implement. Some of the challenges have been identified as Mindset, tools & infrastructure, testing, domain applicability, Understanding, Code dependencies and software requirements

The purpose of Continuous Deployment is to achieve the ability to be able to deploy code to the customers as soon as new code has been produced[3]. This is demanding and challenges for being able to implement such a process have been identified. Some of the challenges are similar to the challenges for

implementing Continuous Integration. The challenges have been grouped into two groups, technical adoption challenges and social adoption challenges.

The path to Continuous Deployment has been described in a model, as the Stairway to heaven, but the task is not trivial to complete and contains barriers[2]. Software organisations evolve practices and strategies over time. Once an agile practice has been implemented a transition towards continuous integration can start. When this step is completed a strategy for implementing a Continuous Delivery process can start to take place before reaching the final state as an R&D experiment system.

Joining a Continuous Delivery process and culture can be frightening in the beginning[6]. But with the help and explanation of more experienced software developers this can be overcome easily and fast. First impression was that it was a bad idea to deliver out to production but once the process was explained the employee showed a more positive attitude towards it.

Organisations are continuing to invest in Continuous Delivery process solutions due to the expected benefits[8]. The expectations are high and by shortening the release cycles an expected outcome is to be able to verify the value for the customer.

Faster feedback by more frequent releases to reduce costs is an important motor in this change[4]. But once again the importance of delivering software with a value for the customer is mentioned.

## 2.1 Definitions extracted from literature

To make the reading of this thesis more understandable from our point of view, we have extracted some interpretations related to the literature we have read on the subject.

**Continuous Integration** aims to integrate each commit to the master branch of the project. The outcome of a successful integration is the latest successful build. In a Continuous Integration process, the aim is not to have release ready code, the aim is to integrate often to avoid integration problems.

**Continuous Deployment** works as an extension of Continuous Integration, instead of leaving the integrated code as the latest build, Continuous Deployment aims to deploy the build to a target. An importance difference from Continuous Integration is that this extension demands that the software is ready for targets or customers, since it will be deployed. This put slightly different requirements at the software, internal experiments should be handled carefully and features not in use should not be visible.

**Continuous Delivery** aims to deliver working software continuously. It doesn't need to include the deployment of software, but it should be ready for a deployment at any given time.

### 3 Methodology

For this specific study we used a Survey study as methodology. We decided on Survey research because we wanted to collect equal data from the sample and have the data interpreted comparatively. To compare trends, attitudes and opinions was also of interest and this made a Survey study suitable.

#### 3.1 Data collection

The data was collected by following a questionnaire during the interviews, which is attached in the appendix. First we studied the literature related to our research questions. We then created questions and answers in an iterative manner consulting software engineering literature on research, Shull et al. [9].

Once the questionnaire was created, it was first tested by two pilots without low or no agile understanding, active within the industry. The idea was to try the questions and how clear they would be, for someone who knew little about the subject. Once the iteration was done and questions adjusted to be more clear, the questionnaire was yet again sent out to pilots. This time three students with good understanding of the agile ways of working.

The questionnaire contained both open-ended and closed-ended questions. The subjects were interviewed at one point in time, cross-sectional approach, i.e. not analysing Continuous Delivery over time. We used two different strategies during our one-to-one interviews.

1. One researcher asked the questions and one researcher wrote down the subjects answer.
2. If only one researcher could participate the answers were voice recorded after asking the subject for permission.

All subjects were asked if they wanted a copy of what was written down during the interview for approval and all of them were interviewed in their natural setting. Transcriptions and audio recordings were stored privately online available for both researchers.

**Population description and sample :** As population we defined Developers, Testers, Enterprise Architects, Software Team Leaders and Managers. We used convenience sampling to contact the population, our own and our supervisors connections to get in contact with the organisations of interest. This can to some extent also be explained in a way that some of our contacts re-directed us to more appropriate subjects, at least when they felt that they were the wrong target.



**Survey and survey structure :** We wanted our analysis to represent the population rather than the sample after completion, which is also discussed by Shull et al. [9]. For this reason a questionnaire was used as our survey instrument, instead of semi-structured interviews, which could give an more personal reflection on the result.

The questionnaire contained three parts. Part one aimed to gather some general data about the subjects professional background and attitude towards Agile and Lean Software Development.

The second part focused on research question number one, where we gathered understandings of Continuous Delivery. We began with asking questions related to how they integrate code during different development projects and then moved on to more general questions.

The third part focused on research question number two. We started with asking questions regarding how they test software and how they would like to improve their testing procedure. We then moved on to more general questions regarding testing in an Continuous Delivery process.

### 3.2 Data analysis

One researcher used a technique where he printed out all raw transcriptions from the interviews and compared them by placing three transcriptions next to each other, this while taking notes and identifying similarities until all transcriptions were studied. The researcher then compiled the notes into a readable format before comparing each transcription once again with the text to ensure validity. The method used had similarities with cross-case analysis discussed by Seamen[11].

The other researcher used a technique where he printed out a compiled collection of all answers. He then read all answers regarding one question at a time and used a type of color coding, using pens to mark answers of interest. This method had similarities with coding also discussed by Seamen [11]. When this was done research findings was moved to a document.

## 4 Results

This section contains three subsections, one for each part of the questionnaire, each subsection lists a summarised answer of respective question in the questionnaire. In total 14 interviews were completed, we reached out to 28 subjects giving us an response rate of 53.5%. Seven out of these were conducted by two researchers, while seven were conducted by one researcher and therefore the sessions were audio recorded. A 15th interview was discarded due to a violation against our methodology. Only one researcher carried out the interview but was not allowed to record the interview.

### 4.1 Questionnaire Section: Info

As stated in the previous section, it was of importance that the subject were well aware of the process used within the organisation they represented in order

to get accurate results. A few background questions were asked initially.

*What is your current position?*

We did manage to reach out to a large variety of the population that was defined. Our interviews covered a wide spectrum of different roles within organisations managers, developers, software engineers, enterprise architects were among the subjects interviewed.

*What is your educational background?*

Over 75% of the subjects had at least a bachelor degree and 50% had a master degree. All degrees were within technical fields.

*For how many years have you been working within the current industry?*

In average the subjects had roughly eleven years of experience within the IT industry.

*For how long have you been working in your current position?*

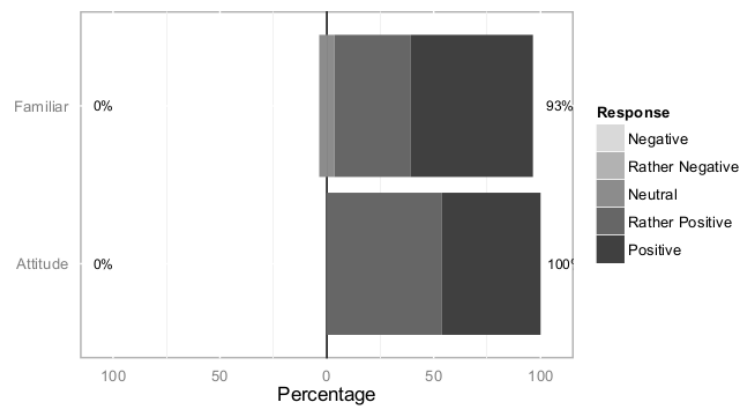
The average time at the current position, were slightly lower than four years.

*To what extent are you familiar with Agile & Lean Software Development?*  
(fig. 1)

The positions our subjects had at the organisation varied. Yet, they all claimed to have at least a good understanding of Agile & Lean Software Development.

*Are you positive or negative towards Agile & Lean Software Development?*  
(fig. 1)

Their attitude was also very positive towards this way of working, no matter their current position within the organisation. While developers appreciated how smooth it could be, the more organisational roles saw different benefits of using an agile approach.



*fig. 1*

#### 4.2 Questionnaire Section: Understanding of Continuous Delivery

The second section of the questionnaire aims to find out more information about the view at Continuous Delivery.

*Do you agree with the following statement regarding your current projects?*

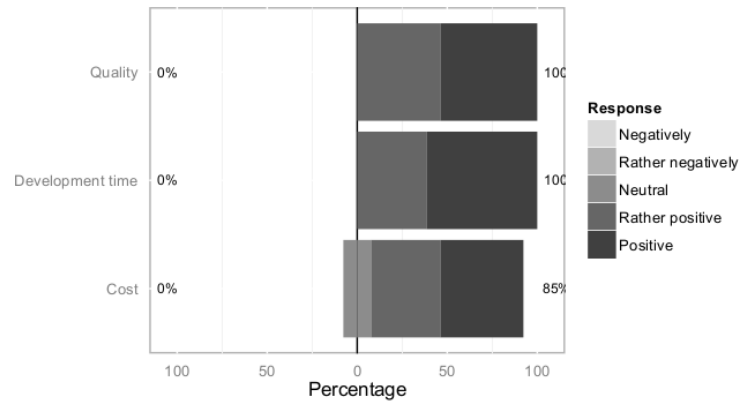
When asked how they currently integrated most of the work within their projects, the subjects revealed that a majority used “Feature boxed” integration plans. The interpretations of “Feature boxed integration” is the use of feature branches. Many organisations use both continuous integration and feature branches to describe their process towards continuous integration. The time boxed integration plans were used, but not commonly, in favor to feature boxed. Big bang integration plans does not exist at all within these organisations.

*If following another integration plan, please write below.*

No subjects mentioned any other integration plans during our interviews.

*How do you think that Continuous Delivery would affect: development time, quality, costs? (fig. 2)*

Moving towards Continuous Delivery is considered in this questionnaire to have a positive effects on the project variables. A majority think that it have a positive effect on the development time, while all subjects were convinced that the effects are at least rather positive. A majority were also convinced that it have positive effects of the quality of the products and that the costs, over time, decreases. Some subjects agreed the short term costs were high, but the long term costs were lower.

*fig. 2*

*What would be the greatest obstacle if/when establishing a Continuous Delivery process?*

When asked for the obstacles to adopt the process, tools and technical challenges were mentioned as the largest obstacles. A number of tools need to work together in order to have a deployment pipeline. Furthermore, current processes and organisational issues is also considered to be large obstacle, examples of existing processes that are not flexible for changes were mentioned, as well as an attitude to “fix all problems with a new process”, an attitude that instead could lead to new problems. Mindset and organisation culture was also mentioned and is closely related to the organisational issues.

*What would be potential benefits of a Continuous Delivery implementation for your organisation?*

When asked for closer details, a majority of the subjects answered that higher quality of the product is the greatest benefit of a process implementation. The possibility to receive and respond quickly to feedback was considered as a major benefits. Increasing the efficiency of the work was also mentioned by the subjects. It was mentioned that the efficiency is not always directly related to the development time, but could also be how well products are developed.

*What would be potential consequences of a Continuous Delivery implementation for your organisation?*

Possible consequences of an implementation are viewed as positive, a majority mentioned the benefits of quality, fast feedback and efficiency as positive consequences as well. The bar to implement the process can be high, but over time all subjects saw it as a good investment for their organisation.

*What would be the potential challenges of implementing Continuous Delivery at your organisation?*

When it comes to implementing Continuous Delivery in the organisation, the mindset among employees and work culture were pointed out as the largest challenge. Also, testing were mentioned as a large challenge, in particular a reliable and working test pipeline. Also to set up this pipeline was considered to be a great challenge, configuration and setup of the tools are considered to be hard to get working.

*How do you think Continuous Delivery would affect your relation with your customers?*

The subjects are satisfied with the concept of Continuous Delivery, that there will always be a delivery ready for their customers and they are able to react to any feedback given. However, it's uncertain to some organisations what the customers attitude to Continuous Delivery is. A majority agreed that Continuous Delivery is good for their relations with the customers, but to take it to the next step, Continuous Deployment, is a step to far, for several reasons. Continuous Deployment may put to much pressure at a customer and damage the relation if the customer is not ready for continuous updates. A mitigation strategy that was mentioned by several subjects was the ability to hide a unfinished feature

for the customers, for example by using a feature toggle.

*How do you think Continuous Delivery would affect your employees interaction with the production code?*

A majority of the subjects agreed that the process have a positive impact at their employees. Positive effects such as: deeper understanding of code and products, awareness of details of what the organisations are doing and the directions that they are heading for. It's believed that more engaged co-workers have a direct effect at the products quality, both the final products and during development. However, it also could put higher demands at the developers, such as to be responsible for the code and trust the system, most subjects agreed that even in that case, the overall impact was in general positive.

*How do you think working with Continuous Delivery would affect the way you outsource/hire third party organisations and communicate with these?*

The subjects had in general a positive attitude towards the impact at outsourcing or hiring services from third party organisations. They thought that it could be slightly harder to integrate outsourcing into their workflow, but as long as the communication is working most had a positive attitude. When it comes to third party services, no changes or effects are believed to happen, our subjects had a positive attitude towards this.

*Is Continuous Delivery something you would like to have established in your organisation? (fig. 3)*

With no doubt, all subjects had a positive eye towards Continuous Delivery and wanted a working process in their organisation.

#### 4.3 Questionnaire Section: Testing

The third and last part of the questionnaire covered a range of questions regarding tests and testing.

*To what extent do you use: Unit tests, Acceptance tests, Performance tests and Manual testing? (fig. 4)*

When it comes to testing, a majority of the subjects witnessed of that they still had a substantial amount of manual testing compared to other kinds of tests. Most organisations used some sort of Unit and acceptance testing. Only a few had no performance tests at all.

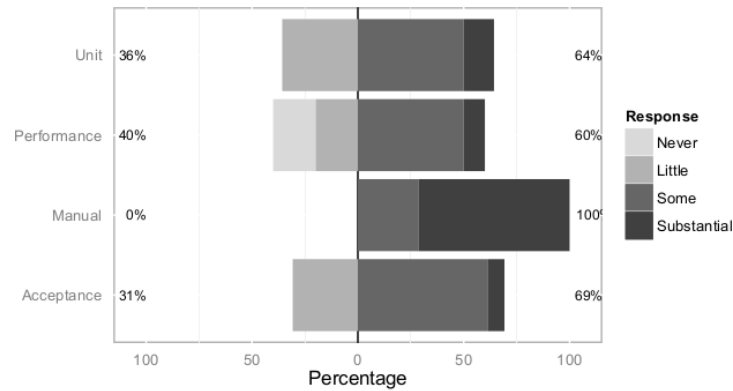


fig. 4

*How reliable would you grade the testing procedure at your company?(fig. 5)*  
 Even though no organisation was satisfied with their current procedure, a majority stated it was reliable.

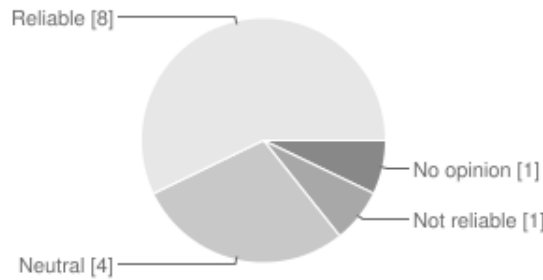


fig. 5

*How would you like to improve your testing procedure?*  
 More automatisation was the most desired improvement of the current testing procedure, as well as improving the current tests. No subjects were satisfied, or close to satisfied, with their current procedure.

*Could you give us an example of something you would like to automate in your testing procedure?*

When asked for a concrete example of something that the subject would have automated, many asked for automatisation of User Interface(U.I) tests. Most subjects agreed that this would be the hardest part to automate for various reasons; many different devices/platforms, “computers can not decide what’s good or bad”, “computers do not have a feeling for looks or appearance”. Within products with no or little user interfaces, there was a demand to run black box/acceptance tests of the systems.

*Is manual testing suitable in a Continuous Delivery Process?(fig. 6)*  
 When asked if manual testing had a role within Continuous Delivery, the sub-

jects were divided. The organisations working with pure software had a majority that disagreed with the statement. While the organisations that had more hardware involved in their development seemed to agree. However, in relation to the previous question, some of the pure software organisations in larger extent stated that U.I was hard, if not impossible, for a computer to test properly.

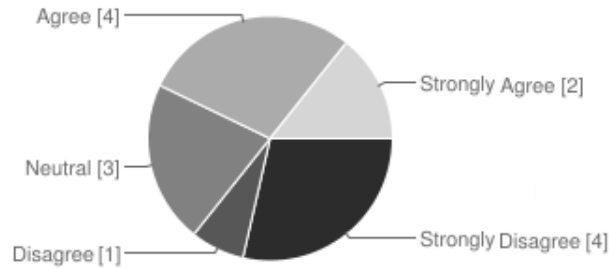


fig. 6

*What do you think are the main obstacles for not being able to automate the complete testing procedure in a company?*

The obstacles for not being able to automate the entire test procedure in an organisation was mostly related to organisational and technical issues. Time and money were stated as main factors for organisational issues, while complexity, tools, technical dependencies was the technical issues. A majority also expressed a sort of scepticism of computers handling all testing, statements as “it will be hard to remove humans completely” was mentioned.

*What is your opinion on small release, small bug?*

A majority agreed with the statement “Small release, small bug”, but two answers was sticking out: “Sometimes our small releases created very large bugs that broke the product” and “In general, it should be reformulated as: small release, small risk”.

*What is your opinion on releasing software often to learn more about the software quicker?*

All organisations agreed with the second statement. The subjects witnessed that the customers are the ultimate testers and the shorter feedback loops that Continuous Delivery allows, are to great assistance for the development. New way of testing possible solutions(A/B tests) and changes of the software during the development have been possible through use of this process. All subjects agreed that follow this statement is a positive and healthy way to work, it is allowing a good and dynamic way to develop the products continuously.

*How do you think Continuous Delivery has affected the way we look at testing software?*

A majority of the subjects witnessed that the work towards a Continuous Delivery process had a major impact at their understanding of testing. Many of them moved from looking at testing as something unnecessary, to organisations that

now value testing and motivate their employees to think more about it, the entire mindset have changed for many organisations. Organisations have adapted a TDD alike approach for developing software, today more tests are done faster than before, with higher demands at the tests. Organisations have started to evaluate where humans match into this new phase of working. Computers can test many things faster than humans, computers also have the ability to repeat an exact test over and over again, but they are not able to get “the feeling” if a product is good or bad. Organisations have realised this and moved their human testers to work more within the current teams and focus more at testing parts of the products that a end-user will use, for example - the design of a U.I.

## 5 Analysis and Discussion

The responses in the first section showed us that we succeeded with our goal to get in contact with people within the industry. It also showed us that they were familiar and positive towards Agile & Lean Software Development.

### 5.1 RQ1: How is Continuous Delivery understood in the industry?

Some of the organisations made a clear distinction between Continuous Delivery and Continuous Deployment, these organisations were often more software oriented. Continuous Delivery according to these organisations is when you are able to deliver to the repository when ready, while Continuous Deployment is when you are able to deploy to the target when ready.

The mindset among employees was something several of the subjects mentioned to be either an obstacle or challenge. People need to change their mindset, in some cases, to be able to achieve and work in a Continuous Delivery process. The correct tools and technologies were also mentioned among the difficulties with implementing a Continuous Delivery process.

The ability to get quick feedback do have a positive impact on the relationship with customers even though some pointed out that mistakes do happen, even with excessive testing.

All subjects wanted Continuous Delivery to be established in their organisation and they were also of the belief that costs, development time and quality would be positively influenced by a Continuous Delivery process.

We observed that the deployment of software within these organisations often were time-boxed, while the integration were continuous. While they aimed to get the latest commits into the latest build, the organisations witnessed of several reasons to not deploy these builds to the customers or targets. Among the mentioned reasons we observed a variety, third party review systems, end-users wish for updates or agreements with customers. A majority of the reasons given, is rooted in that the clients were not able to handle a continuous flow of updates.



## 5.2 RQ2: How does the practice of Continuous Delivery affect the testing procedure within the development cycle?

Most subjects mentioned the importance of automated test suits for each build. On the other hand, in many cases they also said that they think manual testing is a valid part of a Continuous Delivery process, due too that there is no other way to completely automatically test some specific details such as user experiences.

Almost all organisations claimed that testing had a major impact on their view of tests, however, at the same time we did notice scepticism against completely automatic tests. Arguments as “it will be hard to remove humans completely” and “computer cant think out of the box” was mentioned among the subjects.

Many answers pointed out that Continuous Delivery have pushed the need to automate tests and builds further then before. Many organisations also mentioned an increased awareness of testing, that the process had given a deeper understanding of the importance of testing within the organisation.

One of the obstacles mentioned to not be able to automate a complete process is that each time a requirement is changed new tests are needed. The answers showed us that manual testing, to a substantial level, is very common. And to a larger extent then before, organisations include the end-users in the test scope, this was also mentioned during the interviews. Furthermore, organisational issues were also common, time and money to automate more testing was a common issue.

All organisations asked for higher quality of their products, but at the same time they shared organisational issues that made it hard to give testing enough of attention. Priorities within the project were often focused around development instead of testing. Also, initiatives to improve tests are easy to initiate, but seems hard to maintain, due the same organisational issues.

## 5.3 Threats to Validity

The categories used for validity threats were inspired from Easterbrook et al. [10].

**External Validity :** The background of our subjects are of high importance and could lead to an incorrect result if it differs from the defined population to a great extent. When the interviews were arranged, we asked specifically for subjects within the field of software development and verified this in the beginning of each interview by a small chat with the subject as well as our background questions. The willingness of subjects to participate could reflect the outcome of the research. We limited this threat by the amount of organisations contacted.

**Internal Validity :** Failure to analyse qualitative data due to researcher bias was avoided by using two different methods to analyse data. Additionally, our

subjects might get tired of answering the questionnaire during the interview. We estimated the interviews to take approximately 30min. The fact that we used convenience sampling could be a threat but it was still a broad range of organisations due to that we used three different sources for gathering samples. I.e. each researcher and supervisor used their connections.

**Construct Validity :** The design of the questionnaire and our theoretical background could reflect the outcome of the research and perhaps also affect it with researcher bias. This was also prevented as good as possible by our iterations during the design period and the pilot tests.

**Reliability Validity :** In order to deliver results that are valid and described correctly, we both tried to attend to the interviews. In cases where one could not attend, the subjects were asked for permission so that we could audio record it. The notes were looked at by both researchers and the recordings were listened to.

## 6 Conclusion and Future Work

For this bachelor thesis we have carried out a survey study using a questionnaire containing both open-ended and closed ended questions to collect data during one-to-one interviews. The study has been carried out in and around Gothenburg, Sweden. We aimed to increase the knowledge in how the industry understands Continuous Delivery and how they think Continuous Delivery is affecting testing of software.

### 6.1 Conclusion

All organisations interviewed were very positive towards implementing a Continuous Delivery process. As challenges and obstacles these organisations identified a number of items, which have been identified in earlier research[3][2] and articles [4]. Some organisations underlined the importance to differ Continuous Delivery and Continuous Deployment, stating that internally, Continuous Delivery is something that they wish and do aim for in order to increase efficiency, while maintaining or raising quality, to a lower cost.

Our observation is that a product or build of a Continuous Integration process does not necessarily need to have a value for the customer, but instead have a value for the internal organisation. These two may not necessarily need to be combined or synchronised, the purpose and state of a product can change over time, and is allowed to do so, in a Continuous Integration process. In difference to this, any initiative of Continuous Delivery, is required to have customer value.

The purpose of Continuous Delivery, from our observations, is to be able to provide a continuous flow of deliveries of the software product. Therefore, the product is required to be in such a state that it can be delivered at any given time. This being stated, it is not required to deploy the product.

While this is true for these organisations, it seems to us as observers that the challenges are moving from organisational, technical and process challenges within the organisations, stated in an earlier research[2] and an article[4], to challenges outside of the organisations. This is a state between Continuous Integration and Continuous Deployment, where they have to wait for their customers to be ready to adopt continuous deployments.

A multiple-case study[2], follow several companies that makes the transition from Continuous Integration to Continuous Deployment. The researchers who conducted the case study observed three main barriers for an organisation to overcome, in order to reach Continuous Deployment. A few of our subjects stated that they have overcome some very similar issues and were able to deploy continuously, but chose not to do so since that their customers were not ready.

The individual reasons to not deploy the software products among these organisations varied, however, these organisations stated that the clients were not ready for a continuous delivery flow of software. This was also found in previous mentioned research[2], where they observed that the surroundings could affect the possibilities to work in an agile approach.

Earlier research[3], also brings up that not all customers are pleased about receiving continuous updates, which is another answer given to why they would not be ready. General project management benefits from Continuous Delivery such as, shorter development time, reduced costs and improved quality was a general understanding of expected improvements in the industry if or when adapting to a Continuous Delivery process.

In the literature, mentioned in the related work section, we found several articles and reports mentioning and defining challenges of adopting continuous approaches, during our interviews, our subjects witnessed about several of these items. We observed three major categories of challenges during the interviews, Organisational, Technical and Process challenges. In an article[4], items of all these three categories were mentioned with similar results as our own, the items in categories are also found in earlier research[2]. Earlier research[3] also develop the challenges further into categories and specify several more, many of these were found among our results.

The industry expressed thoughts that more automatisation is needed during the procedure of testing their software. It is still considered to be hard to automate all tests so the industry do still accept manual testing as a part of Continuous Delivery. Manual testing is also something, which is still carried out to a substantial level within the industry. Continuous Delivery has increased the level of awareness when it comes to testing and it has also raised the question how to automate the testing procedure. The industry shared the idea of the benefits with shorter feedback loops and also thought that small releases contain a smaller risk.

## 6.2 Methodology

It turned out that the coding method, described in our methodology, was more appropriate by quantifying the qualitative data that the open-ended questions

had generated. This turned out to give an overview of all the interview answers at one time. However, the second method that compared three interviews per time, could be used as support and confirmation of the results. The methods did not collide, but instead we had great use of both.

### 6.3 Future Work

Understandings and implementations change over time. This is why it would be interesting to replicate this research again in a near future, measuring how understandings of Continuous Delivery evolve over time, a longitudinal survey approach.

The industry do believe that customers are not ready for Continuous Deployment. It would be of interest to carry out research on Deployment strategies to be able to continue to shorten the feedback loop and delivery software even more frequently. The area cover both great interest for the industry, as well as for the customers.

**Acknowledgements.** We would like to thank the following organisations for participating in our research: Delphi, Ericsson, eTraveli, HiQ, Jeppesen, Meltwater, Opera, Pulsen, Spotify, Viktoria and VTI.

### References

1. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: "Manifesto for Agile Software Development. Agile Alliance", <http://agilemanifesto.org/principles.html> (Retrieved 5 may 2015).
2. Olsson, H., Alahyari, H., Bosch, J.: Climbing the Stairway to Heaven - A Multiple-Case Study Exploring Barriers in the Transition from Agile Development Towards Continuous Deployment of Software, Proceedings of the 38th Euromicro Conference on Software Engineering and Advanced Applications, pp. 392-399, (2012)
3. Claps, G., Berntsson, R., Aurum, A.: ,On the journey to continuous deployment: Technical and social challenges along the way. Information and Software technology, vol 57, pp. 21-32, Jan, (2015)
4. L. Chen.: "Continuous Delivery: Huge Benefits, but Challenges Too", Software IEE, Vol 32, pp. 50-54, (2015)
5. Humble, J., Farley, D.: Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, 1st edn. Addison-Wesley Professional, (2010)
6. S. Neely and S. Stolt.: Continuous delivery? easy! just change everything (well, maybe it is not that easy), Agile Conference, AGILE 2013, pp.121-128, (2013)
7. Dibbiche, A., Diener, M., Berntsson Svensson, R.: Challenges when adopting continuous integration: A case study, Lecture Notes in Computer Science, vol 8892, pp.17-32, (2014)

8. Leppanen, M., Makinen, S., Pagels, M., Eloranta, V., Itkonen, J., Mantyl, M., Mannisto, T.: The Highways and Country Roads to Continuous Deployment, *Software IEEE*, Vol 32, pp.64-72, (2015)
9. Barbara A. Kitchenham and Shari L. Pfleeger.: Personal Opinion Surveys, *Guide to Advanced Empirical Software Engineering*, Eds. Shull, Forrest; Singer, Janice; Sjberg, Dag I. K., pp. 63-92, (2008)
10. Easterbrook, S., Singer, J., Storey, M-A., Damian, D.: Selecting Empirical Methods for Software Engineering, *Guide to Advanced Empirical Software Engineering*, Eds. Shull, Forrest; Singer, Janice; Sjberg, Dag I. K., pp. 285-311, (2008)
11. Seaman, C.:Qualitative Methods in Empirical Studies of Software Engineering, *ieee transactions on software engineering*, vol. 25, no. 4, july/august (1999)

## Appendix

5/18/2015

Thesis Questionnaire

[Edit this form](#)

### Thesis Questionnaire

#### 1.1

What is your current position?

#### 1.2

What is your educational background?

#### 1.3

For how many years have you been working within the current industry?

#### 1.4

For how long have you been working in your current position?

#### 1.5

To what extent are you familiar with Agile & Lean Software Development?

- ☐ Not familiar
- ☐ Heard of it
- ☐ Neutral
- ☐ Quite familiar
- ☐ Familiar

#### 1.6

Are you positive or negative towards Agile and Lean Software Development?

- ☐ Negative
- ☐ Rather negative
- ☐ Neutral
- ☐ Rather positive
- ☐ Positive

[Continue »](#)

33% completed

Powered by

This content is neither created nor endorsed by Google.

<https://docs.google.com/forms/d/1yiTP61wqYZNJsscOyK6d0eUNv1LL00EwqmuZzzP1N7I/formResponse>

1/2

5/18/2015

Thesis Questionnaire

[Edit this form](#)

## Thesis Questionnaire

### Understanding of Continuous Delivery

#### 2.1

Do you agree with the following statement regarding your current project(s)?

	Not applicable	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
Continuous integration plan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Time boxed integration plan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Feature boxed integration plan	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Big Bang integrations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### 2.1.1

If following another integration plan, please write below.

#### 2.2

How do you think that Continuous Delivery would affect:

	Negatively	Rather negatively	Neutral	Rather positive	Positive
Development Time	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Quality	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Costs	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

#### 2.3

What would be the greatest obstacle if/when establishing a Continuous Delivery process ?

#### 2.4

What would be potential benefits of a Continuous Delivery implementation for your organisation?

#### 2.5

What would be potential consequences of a Continuous Delivery implementation for your organisation?

5/18/2015

Thesis Questionnaire

**2.6**

What would be the potential challenges of implementing Continuous Delivery at your organisation?

**2.7**

How do you think Continuous Delivery would affect your relation with your customers?

**2.8**

How do you think Continuous Delivery would affect your employees interaction with the production code?

**2.9**

How do you think working with Continuous Delivery would affect the way you outsource/hire third part companies and communicate with these?

**2.10**

Is Continuous Delivery something you would like to have established in your organisation?

- ☐ Yes  
☐ No  
☐ No opinion

« Back

Continue »

66% completed

Powered by

This content is neither created nor endorsed by Google.

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)



5/18/2015

Thesis Questionnaire

[Edit this form](#)

# Thesis Questionnaire

## Testing

### 3.1

To what extent do you use?

	Not applicable	Never	Little	Some	Substantial
Unit tests	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acceptance tests	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Performance tests	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Manual testing	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### 3.2

How reliable would you grade the testing procedure at your company?

- ☐ No opinion  
☐ Not reliable  
☐ Neutral  
☐ Reliable

#### 3.3.1

How would you like to improve your testing procedure?

#### 3.3.2

Could you give us an example of something you would like to automate in your testing procedure?

### 3.4

Is manual testing suitable in an Continuous Delivery Process?

- ☐ Strongly disagree  
☐ Disagree  
☐ Neutral  
☐ Agree  
☐ Strongly agree

### 3.5

5/18/2015
Thesis Questionnaire

What do you think are the main obstacles for not being able to automate the complete testing procedure in a company?

**3.6**

What is your opinion on "small release, small bug"?

**3.7**

What is your opinion on "releasing software often to learn more about the software quicker"?

**3.8**

How do you think Continuous Delivery has affected the way we look at testing software?

« Back

Submit

100%: You made it.

Never submit passwords through Google Forms.

Powered by

This content is neither created nor endorsed by Google.  
[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)